CS6460 project catalog

Author: Jos Visser <josvisser66@gmail.com>
Version: 1.0a
Date: 7/19/2020

This document outlines the content of jvisser7-project.zip and instructions for building and using NITS.

Running NITS.

- The executable has been written in Go and can be run on compatible platforms
    - Binaries for Linux and MacOS have been provided:
        - nits-linux
        - nits-darwin
- Run the binary per appropriate way for:
    - E.g: ./nits-linux
- In order to run correctly, NITS needs the “trainhmm” binary (part of standard-bkt; https://iedms.github.io/standard-bkt/).
    - Binaries have been provided for Linux and MacOS.
    - The binary is found in either the following locations:
        - The same directory as the “nits” binary, with the “-<os>” suffix, or
        - $HOME/standard-bkt/trainhmm
    - If you run NITS as indicated above it should find the trainhmm binary.

Using NITS

- NITS has a very simple text based UI. It should speak for itself.
- ? gives help
- There is a debug command that can be used to get information about the internals of nits.
    - The dot command requires the presence of Graphviz on the system.
        - If it isn’t there, a valid dot file will still be generated, but turning it into a PDF will fail.
    - If GraphViz is installed a PDF will be generated called “/tmp/aap.pdf”.
    - If you are running on a Mac, the PDF will be displayed using the Preview application.
- To handpick a question:
    - debug questions
        - To show the question list.

    - debug next <shortname>
        - To select the next question.
    - abandon
        - To abandon the current question and move to the next one.

NITS source code

- The source code is in “main.go” and the Go files in the content/ and nits/ directories.

Compiling NITS

- If you want to compile NITS yourself:
    - Building NITS requires Go 1.14 or higher, which can be obtained on http://golang.org.
    - NITS uses a readline library, which can be obtained as follows:
        - go get github.com/chzyer/readline
    - Once this is in place:
        - go build main.go
        - The resulting binary will be called “main”